

# DS

## تراكيب البيانات DATA STRUCTURES



**ITGS220**

موضوع الدراسة : تراكيب البيانات الخطية Linear data structures

1. القوائم Lists .

▪ ( الطابور الدائري Circular Queue )

أستاذة المادة  
أ. وفاء حسين المصباحي

# DS



## المواضيع التي سيتم دراستها في مقرر تراكيب البيانات

1. مقدمة تمهيدية . Introductory Review

2. تراكيب البيانات الخطية . Linear data structures

3. الترتيب . Sorting

4. تراكيب البيانات الغير خطية . Non-Linear data structures



2. تراكييب البيانات الخطية Linear data structures .

## 2. تراكيب البيانات الخطية Linear data structures :



### □ القوائم Lists .

### ▪ الطابور الدائري Circular queue .

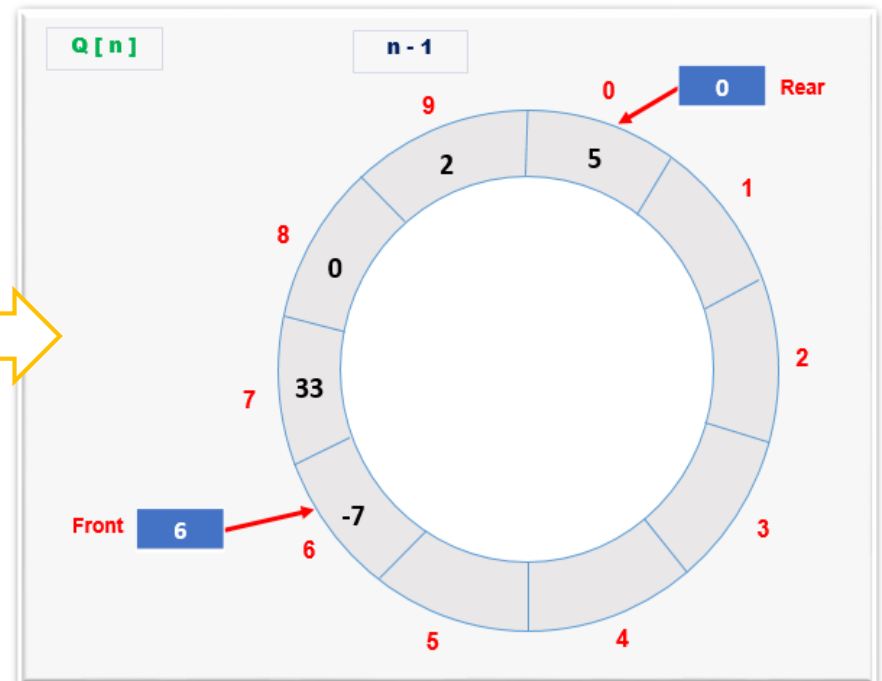
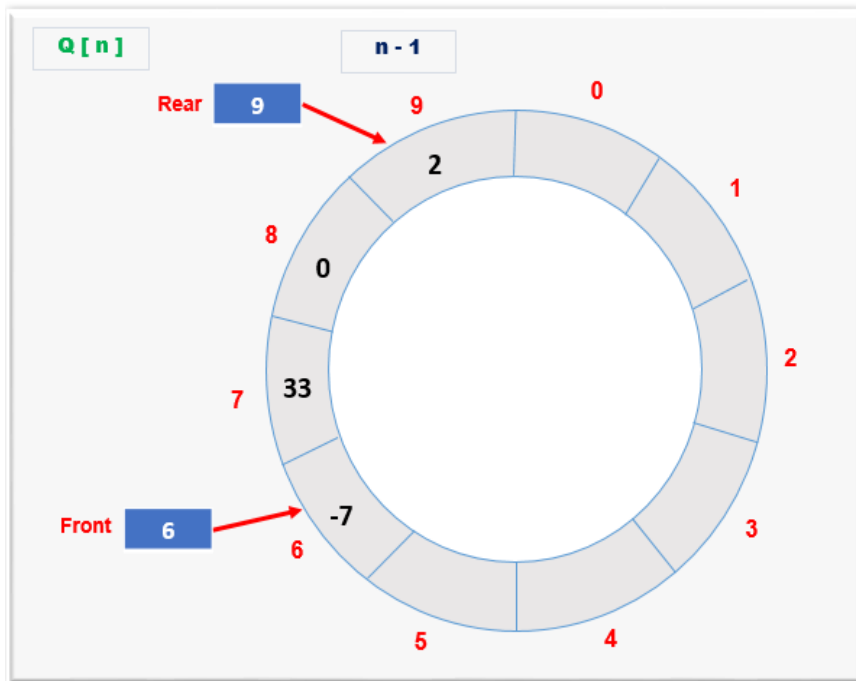
- العمليات التي تجري على الطابور الدائري The operations of circular queue .
- التطبيق للطابور الدائري Implementing of circular queue .
- برنامج بلغة السي لتطبيق الطابور الدائري بواسطة المصفوفة
- Code with C language for implementing a circular queue with an array .



## الطابور الدائري : Circular queue

### التعريف Definition :

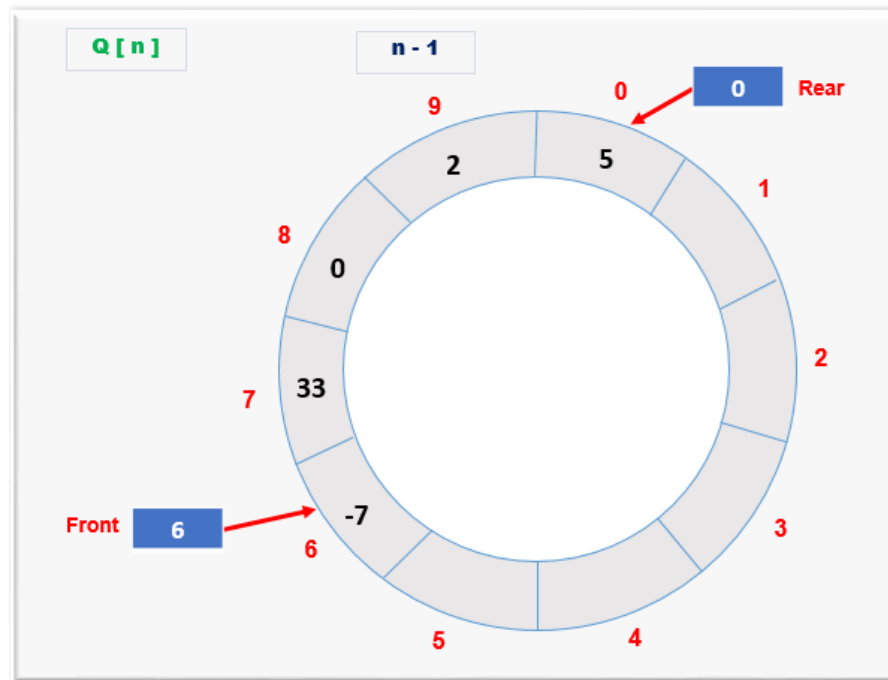
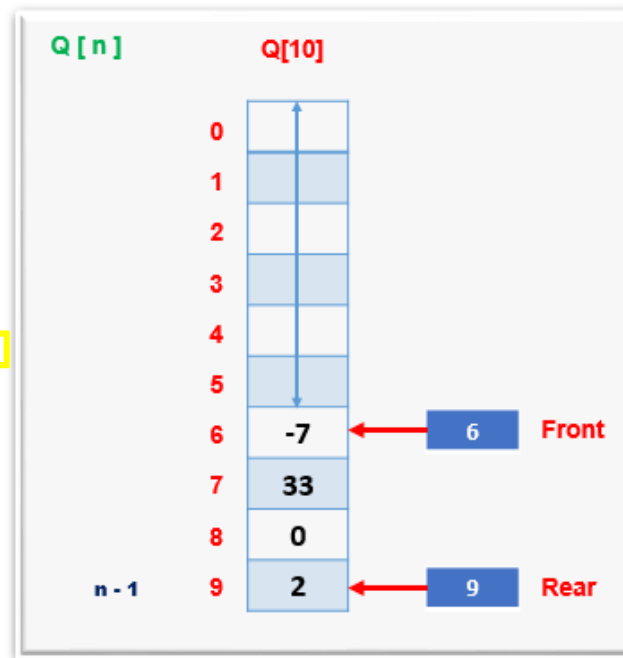
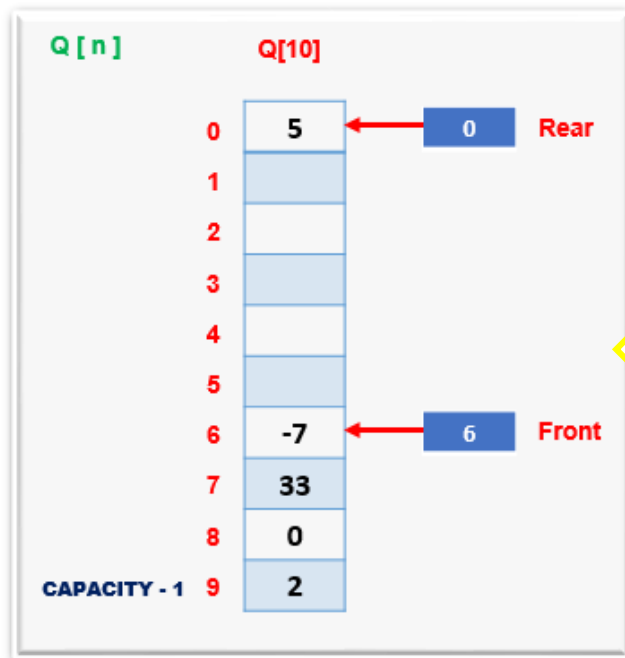
الطابور الدائري circular queue هو الطابور الذي فيه عملية الإدخال insertion لعنصر جديد new item في الموقع الأول first location للطابور إذا الموقع الأخير last location للطابور ممتلئ . الطابور الدائري يوجد له نقطة بداية لكن لا يوجد له نقطة نهاية.





### نفرض:

إذا لدينا طابور لـ  $n$  من العناصر ثم بعد إضافة العنصر في الدليل الأخير ( $n-1$ ) الطابور يبدأ بالدليل 0 ، العنصر التالي يتم إدخاله في الموقع الأول first location للطابور ، وهذا غير ممكن في حالة الطابور الخطي linear queue . لهذا السبب يؤدي الطابور الخطي لسوء استغلال للذاكرة memory وهذا العيب في الطابور الخطي يتم التغلب عليه بواسطة الطابور الدائري .





## العمليات التي تجري على الطابور الدائري : The operations of circular queue

العمليات الرئيسية التي تتجز على الطابور الدائري : circular queue

1. **enqueue(item)** : إدخال insert عنصر في الطابور الدائري circular queue من مؤشر rear .
2. **dequeue()** : إلغاء delete وإرجاع قيمة العنصر الذي يشير له المؤشر front .





## التطبيق للطابور الدائري : Implementation of circular queue

في أغلب لغات البرمجة العالية المستوى high level languages . الطابور الدائري circular queue يمكن أن يطبق بسهولة.

```
CREATE Q[n] , FRONT  $\leftarrow$  -1 , REAR  $\leftarrow$  -1
```

```
INSERT ( Q[n] , REAR , x )  
IF FRONT=0 & REAR = n-1 || FRONT=REAR+1 , THEN "Queue Full"  
IF REAR = -1 & FRONT=-1 , THEN FRONT  $\leftarrow$  0  
IF FRONT>0 & REAR = n-1 , THEN REAR $\leftarrow$ 0 , Q[REAR]  $\leftarrow$  x  
ELSE REAR  $\leftarrow$  REAR + 1 , Q[REAR]  $\leftarrow$  x  
END
```

```
DELETE ( Q[n] , FRONT , REAR )  
IF FRONT = REAR , THEN FRONT = REAR = -1 "Queue Empty"  
IF FRONT = n-1 & REAR < FRONT , THEN FRONT = 0  
ELSE FRONT  $\leftarrow$  FRONT + 1  
END
```

**CREATE**  $Q[n]$  ,  $FRONT \leftarrow -1$  ,  $REAR \leftarrow -1$

**INSERT** (  $Q[n]$  ,  $REAR$  ,  $x$  )

**IF**  $FRONT=0$  &  $REAR = n-1$  ||  $FRONT=REAR+1$  , **THEN** "Queue Full"

**IF**  $REAR = -1$  &  $FRONT=-1$  , **THEN**  $FRONT \leftarrow 0$

**IF**  $FRONT > 0$  &  $REAR = n-1$  , **THEN**  $REAR \leftarrow 0$  ,  $Q[REAR] \leftarrow x$

**ELSE**  $REAR \leftarrow REAR + 1$  ,  $Q[REAR] \leftarrow x$

**END**

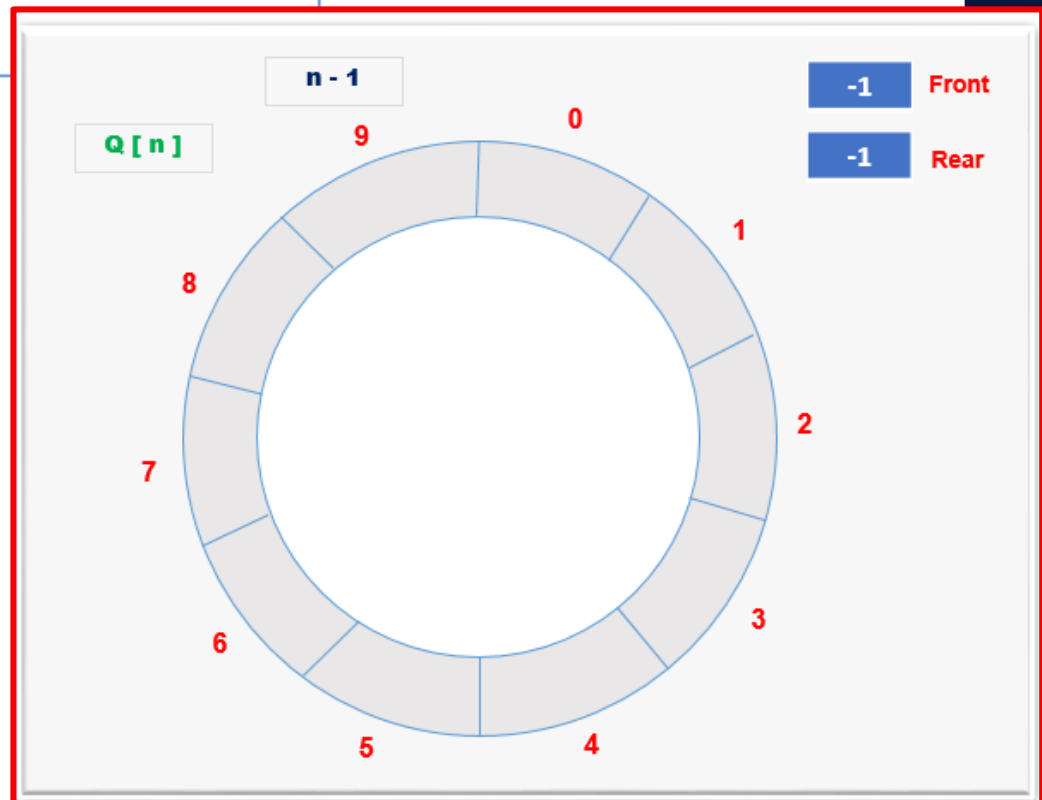
**DELETE** (  $Q[n]$  ,  $FRONT$  ,  $REAR$  )

**IF**  $FRONT = REAR$  , **THEN**  $FRONT = REAR = -1$  "Queue Empty"

**IF**  $FRONT = n-1$  &  $REAR < FRONT$  , **THEN**  $FRONT = 0$

**ELSE**  $FRONT \leftarrow FRONT + 1$

**END**



CREATE  $Q[n]$  ,  $FRONT \leftarrow -1$  ,  $REAR \leftarrow -1$

INSERT (  $Q[n]$  ,  $REAR$  ,  $x$  )

IF  $FRONT=0$  &  $REAR = n-1$  ||  $FRONT=REAR+1$  , THEN "Queue Full"

IF  $REAR = -1$  &  $FRONT=-1$  , THEN  $FRONT \leftarrow 0$

IF  $FRONT > 0$  &  $REAR = n-1$  , THEN  $REAR \leftarrow 0$  ,  $Q[REAR] \leftarrow x$

ELSE  $REAR \leftarrow REAR + 1$  ,  $Q[REAR] \leftarrow x$

END

DELETE (  $Q[n]$  ,  $FRONT$  ,  $REAR$  )

IF  $FRONT = REAR$  , THEN  $FRONT = REAR = -1$  "Queue Empty"

IF  $FRONT = n-1$  &  $REAR < FRONT$  , THEN  $FRONT = 0$

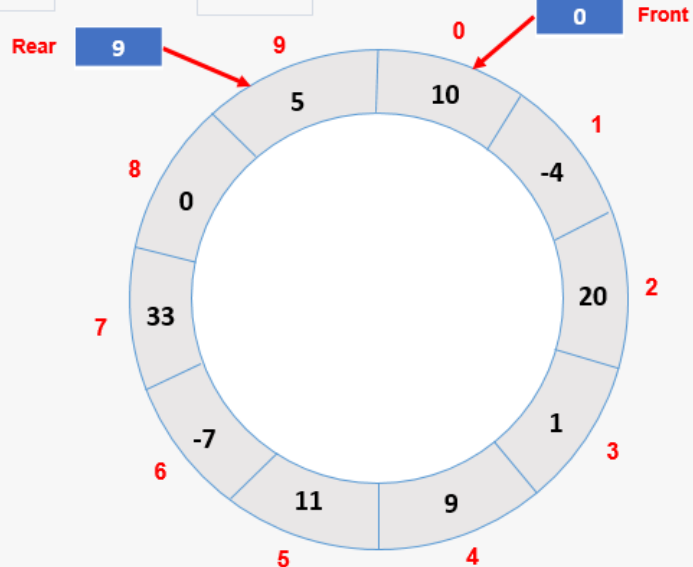
ELSE  $FRONT \leftarrow FRONT + 1$

END

1 2

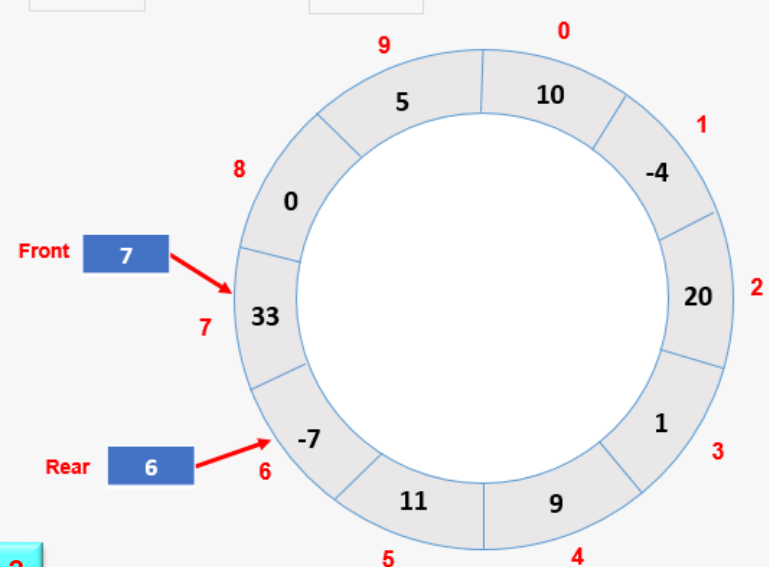
$Q[n]$

$n-1$



$Q[n]$

$n-1$



```
CREATE Q[n] , FRONT  $\leftarrow$  -1 , REAR  $\leftarrow$  -1
```

```
INSERT ( Q[n] , REAR , x )
```

```
IF FRONT=0 & REAR = n-1 || FRONT=REAR+1 . THEN "Queue Full"
```

```
IF REAR = -1 & FRONT=-1 , THEN FRONT  $\leftarrow$  0
```

```
IF FRONT>0 & REAR = n-1 , THEN REAR $\leftarrow$ 0 , Q[REAR]  $\leftarrow$  x
```

```
ELSE REAR  $\leftarrow$  REAR + 1 , Q[REAR]  $\leftarrow$  x
```

```
END
```

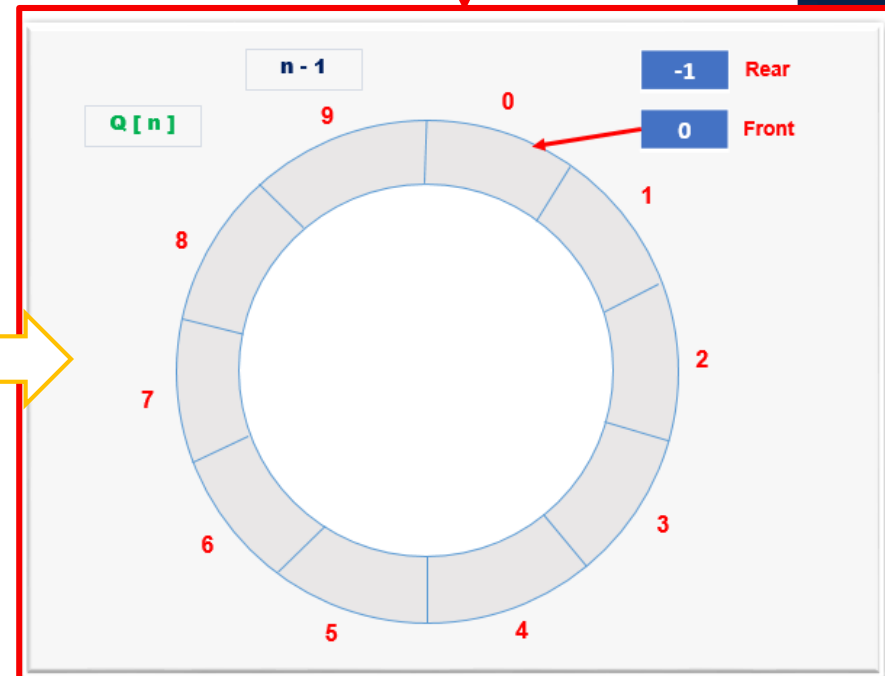
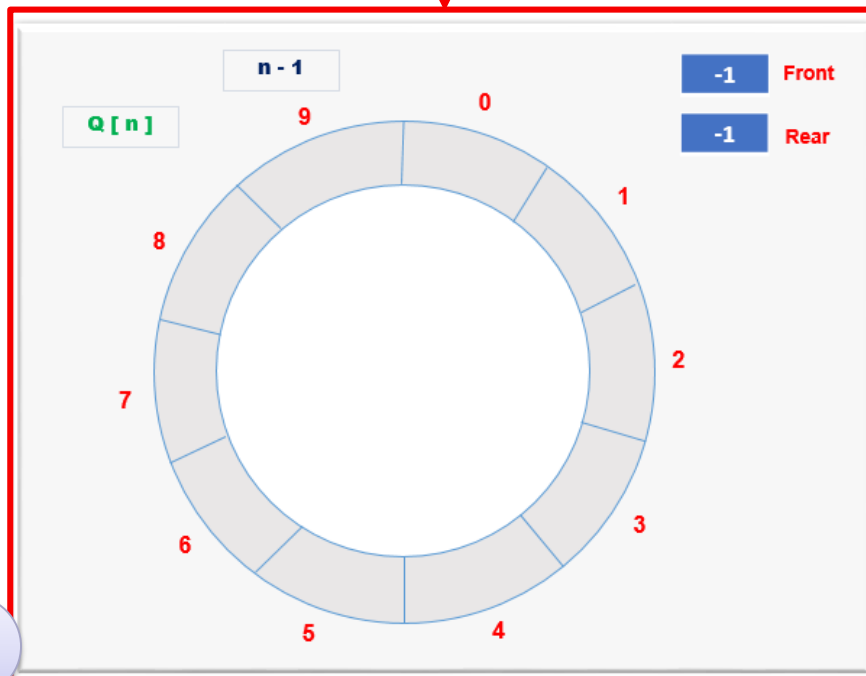
```
DELETE ( Q[n] , FRONT , REAR )
```

```
IF FRONT = REAR , THEN FRONT = REAR = -1 "Queue Empty"
```

```
IF FRONT = n-1 & REAR < FRONT , THEN FRONT = 0
```

```
ELSE FRONT  $\leftarrow$  FRONT + 1
```

```
END
```



```
CREATE Q[n] , FRONT  $\leftarrow$  -1 , REAR  $\leftarrow$  -1
```

```
INSERT ( Q[n] , REAR , x )
```

```
IF FRONT=0 & REAR = n-1 || FRONT=REAR+1 , THEN "Queue Full"
```

```
IF REAR = -1 & FRONT=-1 , THEN FRONT  $\leftarrow$  0
```

```
IF FRONT>0 & REAR = n-1 , THEN REAR $\leftarrow$ 0 , Q[REAR]  $\leftarrow$  x
```

```
ELSE REAR  $\leftarrow$  REAR + 1 , Q[REAR]  $\leftarrow$  x
```

```
END
```

```
DELETE ( Q[n] , FRONT , REAR )
```

```
IF FRONT = REAR , THEN FRONT = REAR = -1 "Queue Empty"
```

```
IF FRONT = n-1 & REAR < FRONT , THEN FRONT = 0
```

```
ELSE FRONT  $\leftarrow$  FRONT + 1
```

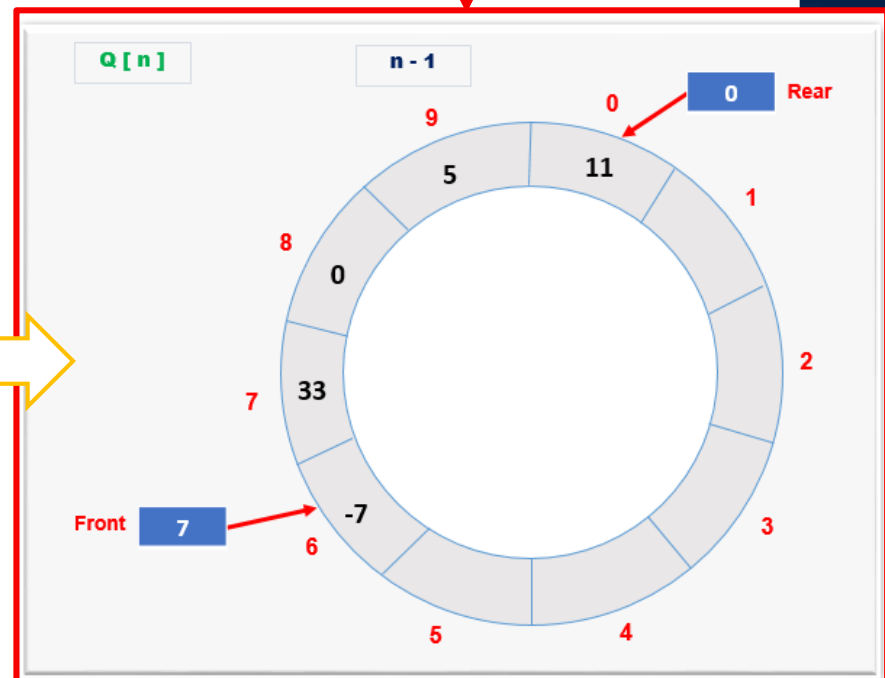
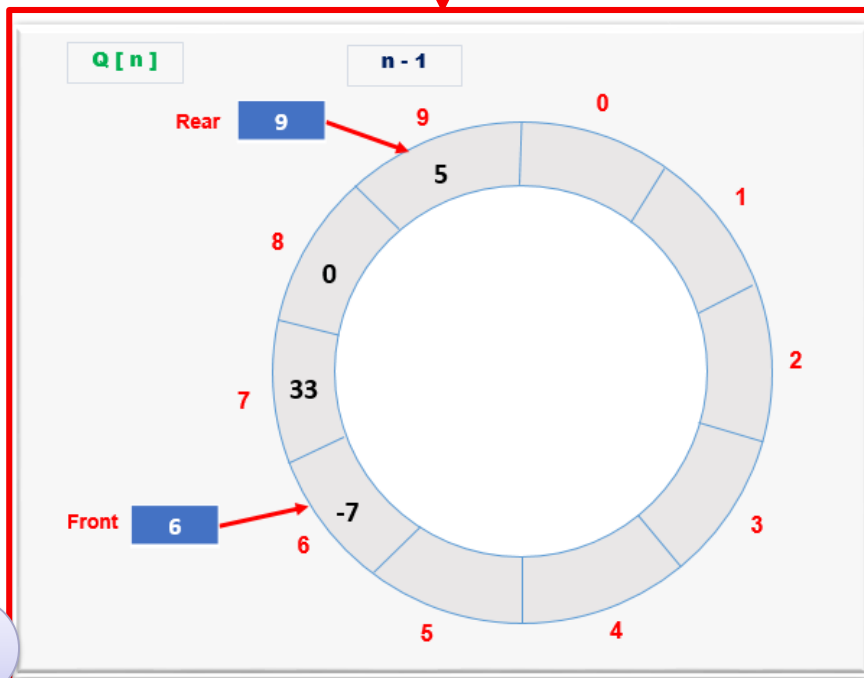
```
END
```



x = 11

REAR=0

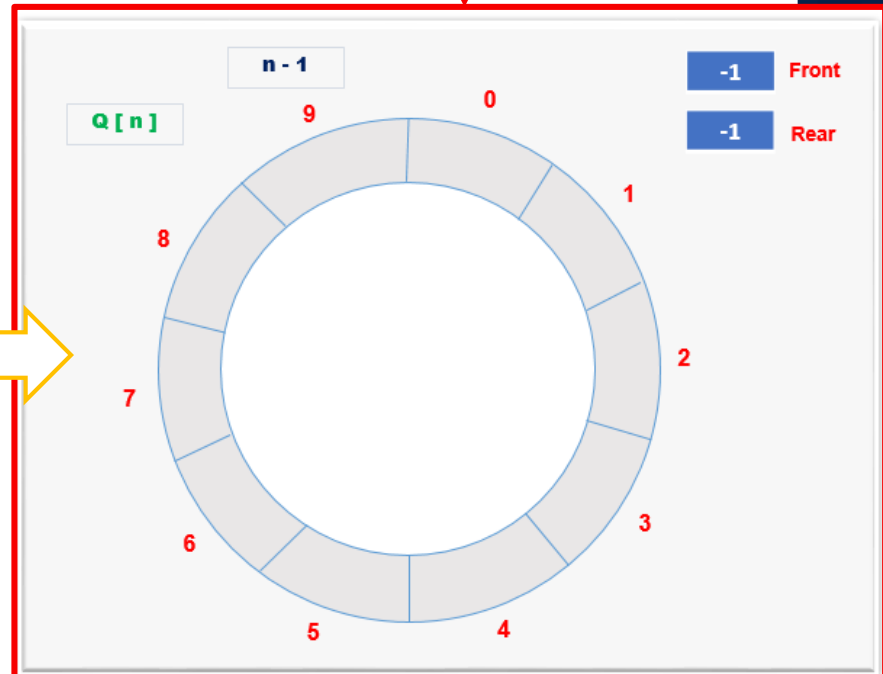
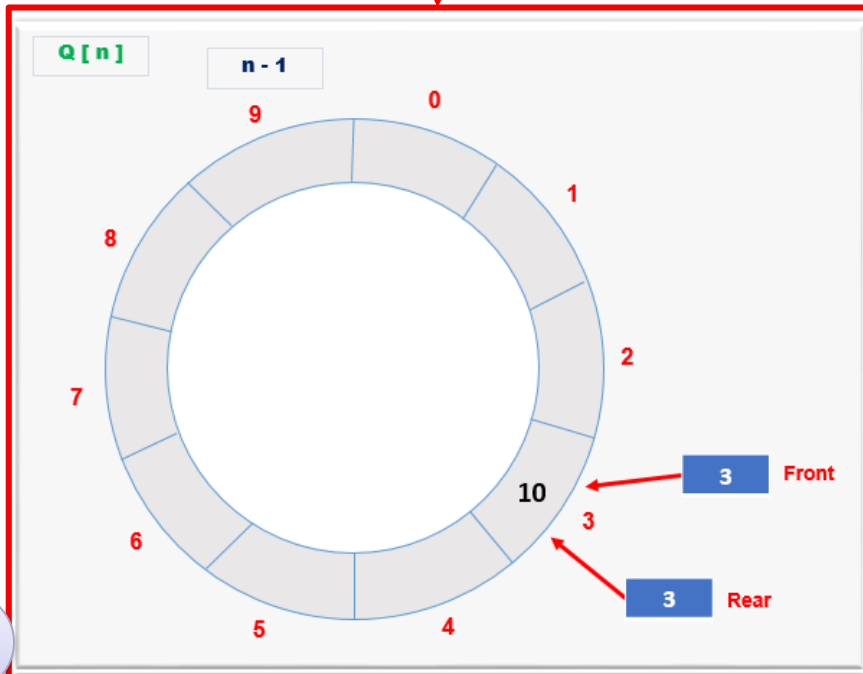
Q [ REAR ] = 11



```
CREATE Q[n] , FRONT  $\leftarrow$  -1 , REAR  $\leftarrow$  -1
```

```
INSERT ( Q[n] , REAR , x )  
IF FRONT=0 & REAR = n-1 || FRONT=REAR+1 , THEN "Queue Full"  
IF REAR = -1 & FRONT=-1 , THEN FRONT  $\leftarrow$  0  
IF FRONT>0 & REAR = n-1 , THEN REAR $\leftarrow$ 0 , Q[REAR]  $\leftarrow$  x  
ELSE REAR  $\leftarrow$  REAR + 1 , Q[REAR]  $\leftarrow$  x  
END
```

```
DELETE ( Q[n] , FRONT , REAR )  
IF FRONT = REAR , THEN FRONT = REAR = -1 "Queue Empty"  
IF FRONT = n-1 & REAR < FRONT , THEN FRONT = 0  
ELSE FRONT  $\leftarrow$  FRONT + 1  
END
```



```
CREATE Q[n] , FRONT  $\leftarrow$  -1 , REAR  $\leftarrow$  -1
```

```
INSERT ( Q[n] , REAR , x )
```

```
IF FRONT=0 & REAR = n-1 || FRONT=REAR+1 , THEN "Queue Full"
```

```
IF REAR = -1 & FRONT=-1 , THEN FRONT  $\leftarrow$  0
```

```
IF FRONT>0 & REAR = n-1 , THEN REAR $\leftarrow$ 0 , Q[REAR]  $\leftarrow$  x
```

```
ELSE REAR  $\leftarrow$  REAR + 1 , Q[REAR]  $\leftarrow$  x
```

```
END
```

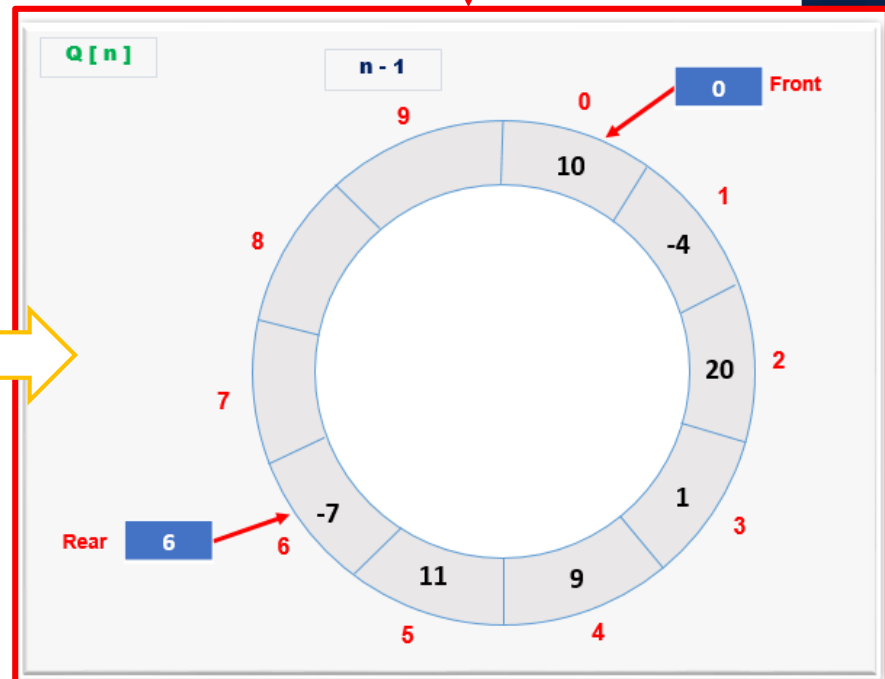
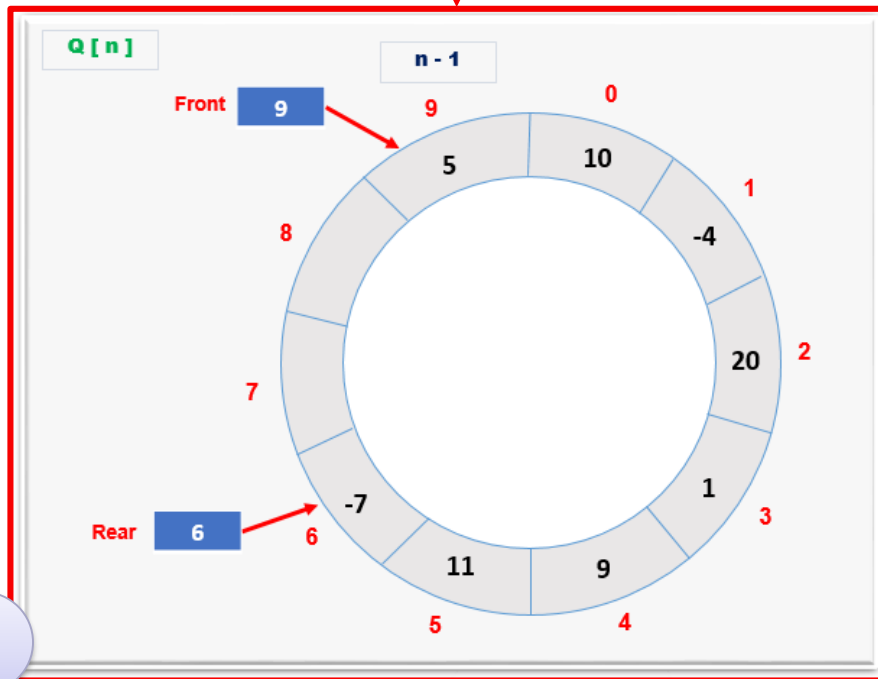
```
DELETE ( Q[n] , FRONT , REAR )
```

```
IF FRONT = REAR . THEN FRONT = REAR = -1 "Queue Empty"
```

```
IF FRONT = n-1 & REAR < FRONT , THEN FRONT = 0
```

```
ELSE FRONT  $\leftarrow$  FRONT + 1
```

```
END
```





**THANK YOU**